

# Towards using the xAPI specification for Learning Analytics in Virtual Reality

Sergej Görzen<sup>1</sup>, Birte Heinemann<sup>1</sup> and Ulrik Schroeder<sup>1</sup>

<sup>1</sup>RWTH Aachen University, Ahornstraße 55, 52074 Aachen, Germany

## Abstract

Virtual Reality (VR) learning applications enable innovative learning opportunities whose effectiveness can be investigated with Learning Analytics (LA). Implementing Learning Analytics in Virtual Reality poses challenges, and isolated solutions are being created. This paper looks at the state-of-the-art current data tracking technologies and presents an approach to facilitate the development process of integrating xAPI for Learning Analytics in VR. It advocates the necessity of restrictions and concepts fostering discourse on additional requirements.

## Keywords

Virtual Reality, Learning Analytics, xAPI, OmiLAXR Framework, Infrastructure

## 1. Introduction

Virtual Reality (VR) applications for educational purposes have garnered significant attention in various research domains, demonstrating positive impacts in educational contexts [1].

The integration of VR technology for educational purposes, considering its multi-modal aspects, presents developers, content creators, and designers with a myriad of challenges spanning diverse hardware setups, didactic and instructional design, software development, and the identification of meaningful metrics for evaluations. Learning Analytics (LA) is emerging as a valuable option for evaluating multi-modal scenarios. With diverse objectives, such as enhancing the learning process, identifying learning behaviors or difficulties, and recommending interventions, LA design is intricately nuanced. The correct tracking of VR activities introduces additional challenges (including the diverse array of VR approaches and equipment) [2]. Developers may struggle not only with the complexities of the LA design process but also contend with challenges related to multi-modal LA.

One possible approach is to use the four dimensions of the Learning Analytics reference model [3]. A concrete definition of *what* data to track (environment) and *how* to achieve this (method) is needed considering all stakeholders (*who*) and goals (*why*). Further, a "correct" and

---


*Joint Proceedings of LAK 2024 Workshops, co-located with 14th International Conference on Learning Analytics and Knowledge (LAK 2024), Kyoto, Japan, March 18-22, 2024.*

✉ goerzen@cs.rwth-aachen.de (S. Görzen); heinemann@cs.rwth-aachen.de (B. Heinemann); schroeder@cs.rwth-aachen.de (U. Schroeder)

🌐 <https://elearn.rwth-aachen.de/goerzen> (S. Görzen); <https://elearn.rwth-aachen.de/heinemann> (B. Heinemann); <https://elearn.rwth-aachen.de/schroeder> (U. Schroeder)

🆔 0000-0003-3853-2435 (S. Görzen); 0000-0002-7568-0704 (B. Heinemann); 0000-0002-5178-8497 (U. Schroeder)

© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

”complete” integration of LA data could influence further results. That makes this part of the LA design step very important for all use cases.

However, technological standards do already exist (see [4, 5]), yet overcoming interdisciplinary and multi-modal challenges and limitations remains a huge task [6]. This paper describes research focusing on reducing the common work for developers while enabling Learning Analytics for Virtual Reality scenarios. We chose the eXperience API (xAPI) specification for the Learning Analytics data format. According to [6], xAPI has chances for multi-modal contexts. However, the research community needs a consensus on working with it. Thus, we developed a software ecosystem that makes working with xAPI more convenient and consistent. This ecosystem contains a tool set and a framework called OmiLAXR (more in section 4).

As technical needs for such tasks are not well published, this paper aims to contribute to the direction of requirements and challenges for enabling Learning Analytics (especially with xAPI) in educational VR scenarios. To achieve this, we present a concept of our approach using xAPI for Virtual Reality and how we mapped a representation of a VR scenario into the xAPI specification. Further, this paper delivers the first results of a study where the participants used our concept in practice supported by a framework we’ve implemented.

## 2. Using xAPI for Learning Analytics in VR

The eXperience API is organized in the JSON data format. It is designed to collect data from a wide range of experiences. Utilizing xAPI, we articulate actors’ activities (agent or group) through structured statements: an *actor* is doing (*verb*) something (*object/activity*). Augmenting these statements with *xAPI extensions* enables the incorporation of additional details, such as learning scenario specifics (*context extension*), detailed information about the target activity or object (*activity extension*), or supplementary insights into task progress (*result extensions*). Each statement fragment has a URI as a unique identifier and additional description. While working with xAPI it is helpful to use xAPI Registries<sup>1</sup> for statement construction. The xAPI specification was derived from SCORM, formerly designed for Learning Management Systems (LMSs). But with the changes for xAPI, it became freer in terms of use and independent from any platform. [7]

At first glance, xAPI is easy to use, but its freedom makes the usage not trivial for virtual reality. According to [6] xAPI has the potential for multi-modal learning scenarios (and VR is one). However, more specifications and research on how to define statements in detail are needed.

After the definition of *what* interactions to track, developers should know exactly how to design them in the form of an xAPI statement. This leads (in our experience) to questions like: How do we call the activities? How generic or specific shall they be? What further information about the interaction is needed? What extensions do we need, and how do we name them? What value format shall the extension be (e.g. tuple, number, struct, etc.)? Unlike in an LMS, where a mouse click triggers an interaction, there are some special challenges in VR. Developers need to decide when users trigger new activities. This includes (for example) defining when users are *moving* or are *nodding* their head, excluding jittering effects. Further, knowing how to

---

<sup>1</sup><https://xapi.com/registry/>, accessed 23.01.2024

handle time-based sensor information like heart rate is important. Options are, e.g. to translate them into activity-based data, make them a part of an xAPI statement in the form of extensions, or ask if an additional data format is needed. Finally, developers may need more complex statements that refer to each other. All of this is possible using xAPI, and xAPI profiles may help with some of these challenges. However, designing xAPI profiles itself is not an easy task. We decided that the usage and maintenance of xAPI registered may be enough for exploration until a unification is found.

### 3. Related Work

There are established models, specifications, and different frameworks for working with LA (see [5]). For example, in [2], the authors proposed a framework for STEM education in VR with the four dimensions of Technology, Pedagogy, Psychology, and Learning Analytics. All four dimensions are important for our framework. Thus, discussing related work in all disciplines would be fair, but we limit here to a small set focusing on data-gathering.

**Related frameworks and tools** The Unity Experiment Framework (UXF) for observing human behavior in virtual environments is explained in [8]. UXF supports real-time data collection with configurable settings and tools, allowing the integration of sensors like eye tracking and EEG. The framework streamlines experiment development. UXF is beneficial for supporting research setups and enhancing data gathering. On the opposite VRSTK [5] takes a holistic approach to support VR experiment creation in Unity. It provides scripts, components, and tools for VR application development. It features scene replay, data import/export, multiplayer support, and tracking of various elements such as movement, gaze, eyes, game objects, and EEG. The UnityGBLxAPI<sup>2</sup> framework already implements xAPI for Unity, focusing on game-based learning and virtual worlds, especially in K-12 education. The implementation of xAPI is very raw and may end up in typos or inconsistencies.

**Previous work** The *xAPI Registry* was initiated to create conventions in the multidisciplinary use of xAPI. Developers and researchers can propose changes over GitLab or the web interface having the same URL as the IRI of a definition (e.g. <https://xapi.elearn.rwth-aachen.de/definitions/virtualReality/verbs/teleported>). Its definition is written in JSON using a strict folder structure represented also by the IRI path (e.g. `{rootFolder}/definitions/virtualReality/verbs/teleported.json`).

Using this xAPI Registry in web projects may be straightforward, but in VR projects, it is still challenging (see introduction). Besides conceptual challenges, the manual application of the registry can still lead to inconsistencies and coding overhead. As presented in [9], we developed the xAPI Definitions Fetcher Tool for synchronizing the xAPI Registry and VR projects. For a better Unity workflow for this tool, we also created the Unity package "xAPI 4 Unity". Instead of providing verbs by strings and dictionaries, we deploy a strict syntax using developers' IDE (strict types, avoiding typos, showing field and method descriptions, consistency, correct usage). Passing an xAPI verb can be done by calling, e.g. `xAPI_Definitions.virtualReality.verbs.teleported`. This approach helps to ensure consistent xAPI development.

<sup>2</sup><https://github.com/gblxapi/UnityGBLxAPI>, accessed 15.12.2023

When developing the VR application *RePiX VR*, we started to extend the xAPI Registry by VR-related vocabulary for assessing the learning progress. We aimed to collect data that help us to understand overall performance in the scenario, actions taken (activities like button presses), gaze direction (eye tracking), head movement (nodding and shaking), and gain insights into the learning environment itself, see [10].

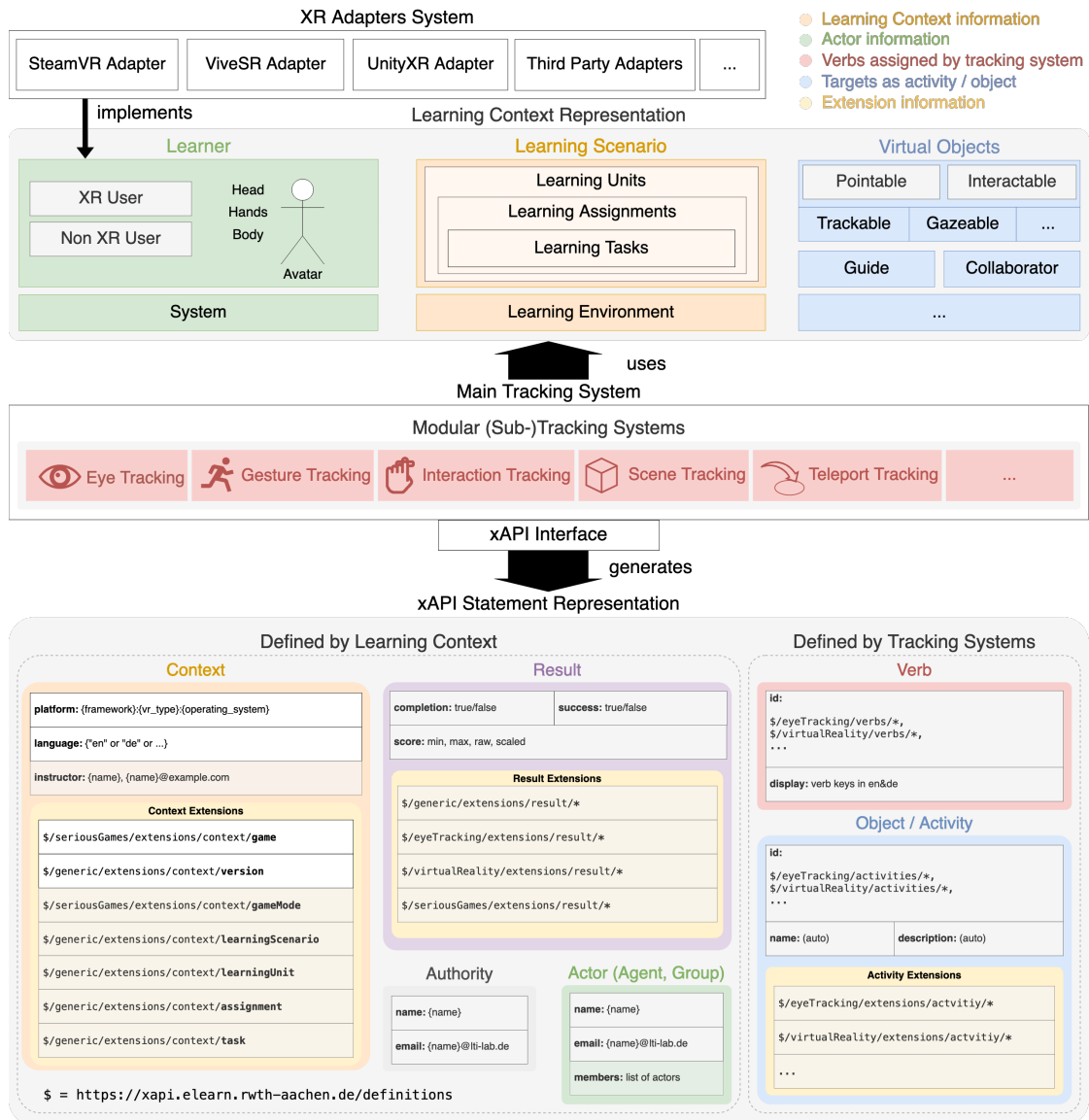
Using a design-based research method, we implemented a data collection library using the strict classes of the xAPI registry specialized for our use case and needs. Based on our experience with this work, we extended the idea to a more generic framework. Considering further technologies, we started to plan in the direction of eXtended Reality (XR). The concepts will be explained in the next section.

## 4. A proposal for supporting LA in XR

To develop a sustainable solution for various applications, one needs to think outside of the scope of a specific application stack. Involving thesis projects, analysis through different dashboards, and conducted studies, we identified additional technical requirements from diverse perspectives. Starting with basic data-gathering components for activities, especially for the learning application *RePiX VR* [10], the result is a dedicated framework, known as *OmiLAXR* (Open and modular integration of Learning Analytics in XR) to facilitate the seamless integration of Learning Analytics in XR applications by using a "Plug & Play"-principle. In Fig. 1, the approach is presented in an abstract representation, aiming to foster discussion while bypassing technical details about Unity. It shows how we have mapped components from a learning context into xAPI.

**XR Adapters System** The *XR Adapters System* is the idea of having a "Plug & Play" mechanism. Adapters can specify through interfaces (for example) how a *XR User* behaviors (teleport, interact, gaze, ...) inside a specific XR framework. Subsequently, these interfaces are utilized to generate xAPI statements by using a model of the current learning context. With the idea of adapters, the integration of third-party libraries is enabled. They are allowed to directly communicate with the *Main Tracking System*, opening the potential to create a bridge between the frameworks mentioned in *Related Work* and the approach for connecting measuring technologies like heartbeat, EEG, etc. Compatibility is an important criterion, and the idea of having adapters may cover it.

**Learning Context Representation** Our concept relies on a comprehensive representation of a learning context (independently from any XR framework), which enriches an xAPI statement with further information. Besides some statements from the *System*, in our design, the statements are generated "Learner-centered". The *Learner* component serves as the primary representation, embodying an XR and desktop representation (*XR User* and *Non-XR User*), including head, body, and hands. This component takes the role of the xAPI *Actor* (highlighted in green). Inside our learning context model, we designed a *Learning Scenario* consisting of multiple *Learning Units*, each containing *Assignments* with numerous *Tasks* (and recursively nested *Sub-Tasks*; highlighted in orange). These components contribute to creating a structured learning path



**Figure 1:** A concept for using xAPI for a VR learning scenario.

within a learning context and need to be defined by developers. In our application, for example, the stages of the rendering pipeline represent learning units, and learners must complete assignments and tasks within each stage to continue the experience of the rendering pipeline (see [10]). In VR, for task completion, there is a need for interaction. As not all *Virtual Objects*, which are distinctly categorized as *Pointables* (for laser pointers) and *Interactables* (for direct interactions), are interesting for analytics, we define the important as *Trackable* or *Gazeable*. Additionally, roles such as *Guide* and *Collaborator*, represented in blue, denote significant player or non-player participants and may potentially serve as xAPI *Activities/Objects*.

**Main Tracking System** The *Main Tracking System* manages a default and extensible collection of modular tracking systems holding a collection of modular tracking controllers handling each a small scope. For example, the *Interaction Tracking System* has separate controllers for laser pointer, mouse, keyboard, or hand interactions.

All tracking events get composed through an xAPI interface to a standardized xAPI format. Final statements get forwarded asynchronously to storage controllers, e.g. for the Learning Record Store, and get caught (in case of connection issues) on the local storage on the hard drive.

**xAPI Statement Representation** As already mentioned, a learning context model is used to set some xAPI fragments (like context, authority, or actor). But verbs and activities have to be defined in (default or custom) tracking systems (including *result-* and *activity extensions*). Following the xAPI specification, a context contains information about the *platform* in free form (in our case, in the format of `{framework}:{vr_type}:{operating_system}`). This records the framework and the VR setup in which the statement was created (e.g. desktop mode and Windows 10). Besides simple information like language and instructor, with the help of context extensions, we added more information about the learning scenario. For example, we track (1) which XR application (*game*) was used, (2) the *version*, (3) if it has a specific *game mode*, and (4) the users location in the learning path (*learning scenario*, *learning unit*, *assignment* and *task*). We defined all needed parts in the xAPI registry (verbs, activities, and extensions for Eye Tracking and Virtual Reality), but we also reused existing contexts, e.g. "seriousGames" or "generic".

## 5. Work in progress and results

As already described, the former concept is an abstract view of what we have realized for applying xAPI for virtual reality as a framework. Thus, an evaluation of this framework is a practical evaluation of this concept. Accordingly, the data-collection mechanism was (pre-)evaluated through smaller studies (see [10]), study courses, and thesis projects for further use cases. Following design-based research cycles, we adapted lacks and extended our framework (e.g. by the learning scenario model) for experiments in learning research. In these studies, the usefulness of the generated xAPI content was validated by creating several explorations. One result, for example, is a Learning Analytics dashboard (see [11]), with which it's possible to compare different variants of a learning scenario. For this, the idea of using context and results extensions of Fig. 1 was used. Each statement contains information at what point (at which task, assignment, and learning unit) it was created and distinguished between different variants of our research object used. In reflection, this concept worked well in conducting research with different variants of the learning scenario. In addition to evaluating the data results, we wanted to evaluate developers' workflow using our approach by finding out challenges and how they are satisfied with the generated statements considering the effort they needed to make. As it is not trivial how to evaluate a framework (and an ecosystem), we created a concept of how to design the study [12] and conducted the study in the summer term of 2023.

Six months, we observed seven computer science master students (with pre-knowledge in LA)

as developers using OmiLAXR for the integration of xAPI into the VR application *Teach-R*<sup>3</sup> and analyzed the OmiLAXR ecosystem according to the guiding criteria: productivity, workflow, usability, functionality, and challenges. As the final results of this observation study are still in progress, we summarize some qualitative results from post-interviews and observations in this paper.

After struggling with setting up the framework, creating xAPI statements was easy and they created statements very easily. The students said they enjoyed having a rich set of existing tracking controllers and welcomed the automatized gathered information. They appreciated creating additional xAPI statements using the xAPI Registry and a strict C# syntax. The students could use xAPI extensions easily but had uncertainties regarding providing the correct types for the values. Overall, they worked with xAPI, knowing the basics without diving deep into specifications.

Using the framework, the students created VR visualizations (e.g. a heat map on a surface). In their use cases, different sources of position data were most important (e.g. position of player, head, and hands). The log of positions is only done on changes and is done on an interval. This behaviour fits well for visualizing the player's path and the heat map, but, e.g. visualizing the head or hands had some jittering effects. This is unambiguously a technical challenge. One challenge was handling system activities (e.g. "System triggered behaviour"). The xAPI specification is not concrete in what an actor is. So, it was allowed to exchange the actor with a system actor. This need was extended in the concept and framework, which happened during the study.

Even though the idea of how to work with xAPI in Virtual Reality is still a work in progress, all goals of the participants for preparing Learning Analytics for Teach-R were completely achieved. Missing components could be added easily due to the modular design. A developer-near representation for xAPI management (and according to mechanisms realized by the framework) helped here well. Diving deep into xAPI specification was unnecessary except for a few basics.

## 6. Discussion

The xAPI specification gives some freedom in how to use it. We consistently observed beginners struggling to use xAPI. In addition, repetitive work often has to be done, and in bigger projects, the usage without any guidelines may result in inconsistencies.

Guided by a framework, the design of how to map a VR scenario into xAPI statements worked well for our use cases. Further, the idea of the *Learning Context Representation* was clear on both sides: our developer participants understood it fast, and also, for analysis by educators, it was useful [11]. But at the same time, the system needed to be explained because of bad documentation. Here, it is important to be careful and find a good balance for instructional designers and developers. The aim shall be to find an easy interface for designing a learning scenario and to generate useful xAPI statements from it.

Further, this approach does not explain how to handle time-based information best, like movement or heart rate, avoiding a huge collection of senseless or repeating data. We think this is clearly a technical challenge that has to be explored more.

---

<sup>3</sup><https://teach-r.de>, accessed 23.02.2024

In addition, this approach focuses on "simple logs" reflecting users' "breadcrumbs" [13] without more complex interdependencies. Although simple metrics may suffice for many analytics goals, incorporating more complex xAPI statements, such as those involving context activities [7] or combining multiple statements semantically to a new one, could be beneficial. For example, the combination of looking at the instruction, nodding with head, and solving the task may lead to a statement like "actor understood the task". To some degree, this can be done in post-analyses, but we believe that more interdependent tasks may open more doors to information and, thus, deeper analytics.

However, while we've delved into specific requirements and presented a concept for our xAPI creation in VR, several questions still exist: What are the specific requirements for LA in VR? Are more complex xAPI statements necessary, and if so, what form should they take, and how can they be implemented? How can we ensure good quality in using xAPI for VR?

## 7. Conclusion

In this paper, we presented an approach to how to use xAPI for VR. This idea was guided by a framework and was evaluated using (directly) two VR scenarios. Implementing xAPI effectively in Virtual Reality is possible but demands constraints and tool support. These constraints could be a specific syntax, a proper xAPI registry, profiles, or a combination of these elements. The presented framework supports the first features. Although the framework is built for Unity, we presented our technical concepts on a more abstract level to make them transferable to other platforms. For example, we plan an exploration of WebXR technology.

Supporting programmers in creating good quality Learning Analytics data is an iterative process that is difficult to evaluate and depends on discussion with others. The effort shows that a modular framework design may support many applications but makes the startup harder.

Nevertheless, xAPI may be suited for VR, but the learning community is confronted with conceptual and technological challenges. We need to explore more use cases and find the limitations of xAPI in Virtual Reality. Focus groups with Virtual Reality and Learning Analytics experts are important to discuss findings and gather valuable input.

## References

- [1] J. Radianti, T. A. Majchrzak, J. Fromm, I. Wohlgenannt, A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda, *Computers & Education* 147 (2020) 103778. doi:10.1016/j.compedu.2019.103778.
- [2] A. Christopoulos, N. Pellas, M.-J. Laakso, A Learning Analytics Theoretical Framework for STEM Education Virtual Reality Applications, *Education Sciences* 10 (2020) 317. doi:10.3390/educsci10110317.
- [3] M. A. Chatti, A. L. Dyckhoff, U. Schroeder, H. Thüs, A reference model for learning analytics, *International Journal of Technology Enhanced Learning* 4 (2012) 318. doi:10/gdm24h.
- [4] S. Schürstedt, A. Geiger, Einsatz von VR-Technologien in BIM/GIS, in: *Proceeding: 31. Forum Bauinformatik*, Universitätsverlag der TU Berlin, Berlin, 2019.



- [5] M. Wolfel, D. Hepperle, C. F. Purps, J. Deuchler, W. Hettmann, Entering a new Dimension in Virtual Reality Research: An Overview of Existing Toolkits, their Features and Challenges, in: 2021 International Conference on Cyberworlds (CW), IEEE, Caen, France, 2021, pp. 180–187. doi:10.1109/CW52790.2021.00038.
- [6] M. Ehlenz, B. Heinemann, T. Leonhardt, R. Röpke, V. Lukarov, U. Schroeder, Eine forschungspraktische Perspektive auf xAPI-Registries, in: DELFI 2020 – Die 18. Fachtagung Bildungstechnologien der Gesellschaft für Informatik e.V., Gesellschaft für Informatik e.V., Bonn, 2020, p. 6.
- [7] B. Miller, Deep Dive: Result, <https://xapi.com/blog/deep-dive-result/>, 2013.
- [8] J. Brookes, M. Warburton, M. Alghadier, M. Mon-Williams, F. Mushtaq, Studying human behavior with virtual reality: The Unity Experiment Framework, *Behavior Research Methods* 52 (2020) 455–463. doi:10.3758/s13428-019-01242-0.
- [9] B. Heinemann, M. Ehlenz, S. Görzen, U. Schroeder, xAPI Made Easy: A Learning Analytics Infrastructure for Interdisciplinary Projects, *International Journal of Online and Biomedical Engineering (iJOE)* 18 (2022) 99–113. doi:10.3991/ijoe.v18i14.35079.
- [10] B. Heinemann, S. Görzen, U. Schroeder, Teaching the basics of computer graphics in virtual reality, *Computers & Graphics* 112 (2023). doi:10.1016/j.cag.2023.03.001.
- [11] B. Heinemann, S. Görzen, A. Dragoljić, L. F. Meiendresch, M. Troll, U. Schroeder, A Learning Analytics Dashboard to Investigate the Influence of Interaction in a VR Learning Application, in: *Learning Analytics for Virtual Reality (LAVR) Workshop at the 14th International Conference on Learning Analytics and Knowledge (LAK24)*, Kyoto, Japan, 2024.
- [12] S. Görzen, B. Heinemann, U. Schroeder, Ein Konzept zur Evaluierung eines Ökosystems für die Integration von Learning Analytics in Virtual Reality, *Gesellschaft für Informatik e.V.*, 2023.
- [13] V. Camilleri, S. de Freitas, M. Montebello, P. McDonagh-Smith, A case study inside virtual worlds: Use of analytics for immersive spaces, in: *Proceedings of the Third International Conference on Learning Analytics and Knowledge - LAK '13*, ACM Press, Leuven, Belgium, 2013, p. 230. doi:10.1145/2460296.2460341.

## A. Online Resources

The application, framework and all sources are available via

- OmiLAXR Website, OmiLAXR Docs for Developers, OmiLAXR: xAPI Data Tracking Ecosystem (GitLab),
- "xAPI 4 Unity" Package, xAPI Definition Fetcher,
- VR Learning Application (RePiX VR).